AMENDMENTS TO THE SPECIFICATION

On page 2, after line 17, please add the following:

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a flow diagram illustrating a securing method according to the invention.

Please replace the paragraph beginning on page 2, line 18, as amended in the amendment filed on February 12, 2009, with the following paragraph:

DESCRIPTION OF THE PREFERRED EMBODIMENTS

According to the invention, the method for securing a computer system by logical confinement of data (see Figure 1) comprises separation of said data per possessor P1...Px...Pn and their encryption with a dedicated key; this separation and encryption process is performed by a procedure comprising the following steps (see figure 1):

- an allocation 11 of memory MAUx performed by a memory manager MM on request from another component of the operating system which transmits to said memory manager MM, the identity Id of the requester Px. This requester will become the possessor Px of the allocated memory MAUx. Transmission of the identity Id of the requester may be accomplished either by managing a current context, or by passing parameters to the functions of the memory manager;

- a check 12 by the aforesaid memory manager of the whole of the memory allocation units MAU1-MAUn, each being associated with a possessor P1-Pn of the memory allocation unit. Each memory allocation unit can only have one single possessor; nevertheless, several memory allocation units may have the same possessor;
- an encryption 13 of the data of each possessor by means of a key associated with this possessor;
- optionally, a use of a secret associated with each possessor, by the memory manager. This secret may typically be provided to the memory manager by the operating system at the moment when the possessor is introduced into the system and upon each access to a memory allocation unit;
- optionally, a use of a key for each possessor by the memory manager. This key may for example be derived from a secret associated with the possessor and a so-called "master" key to which only the memory manager has access;
- a check 14 of the identity Id of the requester by the memory manager for each request to access a memory allocation unit; if this identity Id is not identical with that Idx of the possessor of said memory allocation unit, then the access to the memory allocation unit MAUx is refused 15 by the memory manager;
- performing, by means of the memory manager, encryption 17 (in the case of a write request) or decryption 16 (in the case of a read request) of the

relevant data with the key associated with the possessor, whereby this key may be re-calculated by the memory manager.

On page 2, line 22, please delete the heading and paragraph added in the amendment filed on February 12, 2009.

Please replace the paragraph beginning on page 3, line 30 with the following paragraph:

- this attempt may be triggered via the memory manager: in this case, the check 14 performed by the memory manager automatically leads to rejection 15 of the request;

Please replace the paragraph beginning on page 4, line 28 with the following paragraph:

- The memory manager may also integrate into each memory unit, an area allowing its integrity to be checked, for example from a simple signed checksum or a cryptographic algorithm. The datum contained in this area is updated by the memory manager upon each write access to the unit. It may be used by the memory manager for checking purposes, either systematically at each access to the unit, or periodically. The check 14 before the requested access simply consists of recalculating the integrity datum from the contents of the unit (plain data) and comparing it with the

datum contained in the integrity area. An untimely or illegal change in the contents of the unit may then be detected, which will reinforce security of the data management.